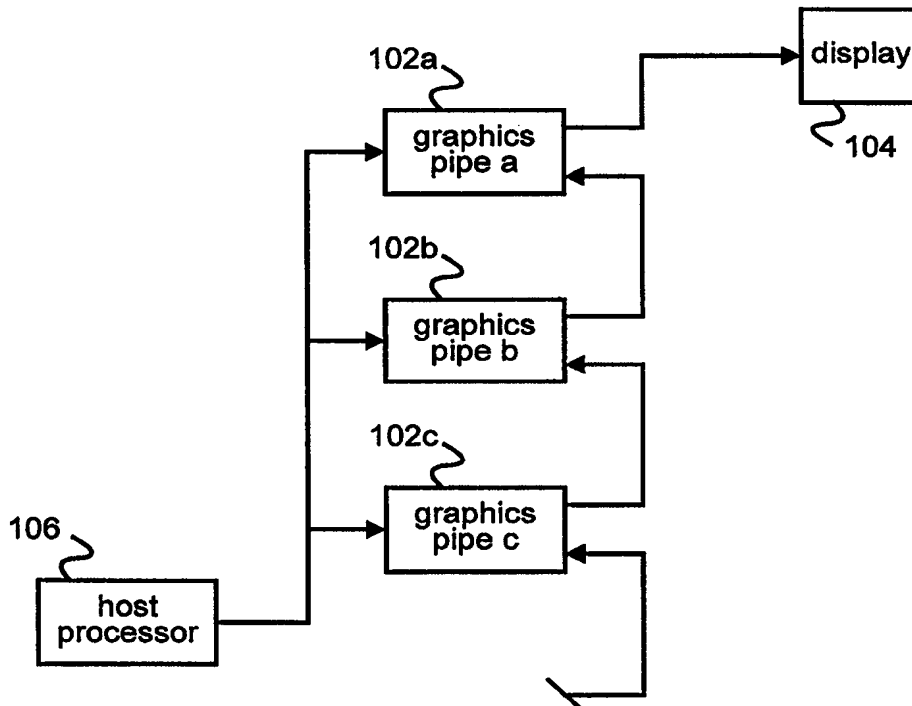**PCT**

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification [7] : G06T 15/00	A1	(11) International Publication Number: **WO 00/41136**
		(43) International Publication Date: 13 July 2000 (13.07.00)

<table>
<tr><td>

(21) International Application Number: PCT/US00/00549

(22) International Filing Date: 7 January 2000 (07.01.00)

(30) Priority Data:
09/227,227   8 January 1999 (08.01.99)   US

(71) Applicant: SILICON GRAPHICS, INCORPORATED [US/US]; 1600 Amphitheatre Parkway, MS–710, Mountain View, CA 94043–1351 (US).

(72) Inventors: BOWEN, Andrew, D.; 5461 Copeland Lane, San Jose, CA 95124 (US). BUCHNER, Gregory, C.; 690 Covington Road, Los Altos, CA 94024 (US). ARNAUD, Remi, Simon, Vincent; 1482 Redmond Avenue, San Jose, CA 95120 (US). CHIAN, Daniel, T.; 1220 Fair Oaks Avenue #312, Sunnyvale, CA 94089 (US). BOWMAN, James; 1946 Pescadero Road, Pescadero, CA 94060 (US).

(74) Agents: SOCKOL, Marc, A. et al.; Graham & James LLP, 600 Hansen Way, Palo Alto, CA 94304–1043 (US).

</td><td>

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

</td></tr>
</table>

(54) Title: METHOD AND APPARATUS FOR SYNCHRONIZING GRAPHICS PIPELINES

(57) Abstract

A method and apparatus for synchronizing the execution of a sequence of graphics pipelines is provided. For a representative embodiment a sequence of graphics pipelines are connected in a daisy–chain sequence. Each pipeline operates can be controlled to operate in one of two modes. The first is a local mode where the pipeline outputs its own digital video data. The second is a pass–through mode where the pipeline outputs digital video data received from preceding graphics pipelines. The pipelines are configured to allow an application executing on a host process to select the next pipeline that will enter local mode operation. The pipeline that is selected to enter local mode operation asserts a local ready signal when it is ready to begin outputting its digital video information. The selected pipeline enters local mode operation. The remaining pipelines in the sequence either enter or remain in pass–through mode operation.

Method and Apparatus for Synchronizing Graphics Pipelines

FIELD OF THE INVENTION

The present invention relates generally to systems for computer graphics. More specifically, the present invention includes a method and apparatus for

5　synchronizing the sequential operation of a group of graphics pipelines.

BACKGROUND OF THE INVENTION

Computer systems (and related devices) typically create three-dimensional images using a sequence of stages known as a graphics pipeline. During early pipeline stages, images are modeled using a mosaic-like approach where each

10　image is composed of a collection of individual points, lines and polygons. These points, lines and polygons are know as primitives and a single image may require thousands, or even millions, of primitives. Each primitive is defined in terms of its shape and location as well as other attributes, such as color and texture.

The graphics pipeline maps each primitive into a memory storage device

15　know as a frame buffer. Each storage location within the frame buffer defines one pixel within the image being produced. The graphics pipeline performs the mapping process by determining which pixels (i.e., which frame buffer storage locations) are included within each primitive. Each pixel is then initializes to reflect the attributes of the primitive, or primitives in which it is included. In many cases, the graphics

20　pipeline will further modify the pixel values in the frame buffer to apply texture, lighting and other effects to the graphics primitives.

The task of rendering primitives to pixels can be very time consuming. This is especially true for complex images that include many primitives or require complex

lighting, shading or other effects. The time consumed transforming primitives becomes problematic for applications, such as flight simulators and virtual reality environments, where rapid image generation is required.

One method for improving the speed of the rendering process is to use
5   multiple graphics pipelines. For this method, the task of rendering primitives is subdivided so that each pipeline performs some portion of the total task. One way of subdividing the rendering task is to assign a fixed portion of each image to a different pipeline. The pipelines operate in parallel and each pipeline processes the primitives that are included in its image portion.

10   The parallel output of the various pipelines is combined for display. To be effective, this combination must be accomplished seamlessly, as though the image were being created by a single graphics pipeline. In general, this seamless integration requires that the output of each pipeline be synchronized. Otherwise, the pipelines produce output at somewhat random intervals. This can make the resulting
15   images appear to be inaccurate or even incoherent.

For some systems, the synchronization problem may be overcome through the use of a global ready signal. SILICON GRAPHICS INC. provides a signal of this type, known as SWAP_READY, for use with certain types of SGI computers. The SWAP_READY signal is physically connected to each graphics pipelines that is to
20   be synchronized. Systems that are ready to output their image portion assert local ready signals. SWAP_READY functions as a logical AND of these local ready signal. As a result, SWAP_READY becomes asserted when all systems are ready to output their image portions. The graphics pipelines sample the state of the SWAP_READY

2

signal on a periodic, synchronized basis. The pipelines output their image portions when the discover that SWAP_READY is high.

A second way to combine multiple graphics pipelines is to subdivide the rendering task so that the pipelines operate in round-robin or sequential fasion. For

5   this method, each pipeline contributes a succeeding image to a sequence of images. The sequential method has certain advantages when compared to the method where each image is partitioned between pipelines. One of these advantages is the ability to use less powerful graphics pipelines to produce a sequence of images at an accelerated rate. This is especially beneficial for flight simulators and other real

10   time environments.

To ensure image quality, sequential pipeline operation requires synchronization of the pipelines that are used to generate an image. Otherwise, there is a chance that the pipelines will output images out of order. For animation and simulation environments, this can cause the display to move unpredictably

15   forward and backward in time. Unfortunately, a synchronization method, such as a global ready signal, is not available for systems using the sequential pipeline operation. As a result, complex software synchronization is required, increasing the difficulty of implementing systems of this type.

For some applications it is desirable to use a combination of both of the

20   previously described techniques for combining multiple graphics pipelines. This allows images to be partitioned into regions with each region using one or more pipelines connected in sequence. The sequential operation of each group of one or more pipelines proceeds in parallel with the operation with the remaining pipeline

groups. This provides a flexible method for arranging graphics pipelines to match the image generation requirements of specific applications. Unfortunately, the lack of a synchronization method for sequential pipeline operation makes combination of these systems with other graphics pipelines difficult.

5   Based on the preceding it may be appreciated that a need exists for an effective method for synchronizing multiple graphics pipelines that are configured to operate in a sequential fashion. A preferable solution to this problem would allow sequential graphics pipeline groups to operate in parallel with other pipeline groups.

SUMMARY OF THE INVENTION

An embodiment of the present invention includes a method and apparatus for synchronizing the operation of graphics pipelines. This method and apparatus allows graphics pipelines to be grouped for sequential operation and allows parallel execution between these pipeline groups. For a representative embodiment of the present invention, one or more graphics pipelines are connected as a daisy-chain sequence. This means that the output of each pipeline (except the last) is connected as an input to a subsequent pipeline. The output of the last pipeline is connected to a display device.

Each pipeline may be configured to operate in two different modes: local and pass-through. Pipelines operating in local mode output their own digital video data (i.e., the pixel values generated by the graphics pipelines). Pipelines operating in pass-through mode output the video data that they receive from preceding pipelines.

In operation, an application program configures the sequence of pipelines so that one pipeline operates in local mode. The remaining pipelines operate in pass-through mode. This causes the digital video output produced by the one pipeline to be passed through to the output device. At periodic intervals, the application program selects the next pipeline for operation in local mode. Shortly after the selected next pipeline becomes ready a swap event occurs. During the swap event, the selected next pipeline switches to local mode and the remaining pipeline either enter or stay in pass-through mode. In this way, the pipelines cooperate in sequential fashion to produce a sequence of images.

The timing of swap events is based on the state of a global ready signal. The global ready signal represents a logical AND of a set of local ready signals. One of these local ready signals is produced by the selected next pipeline. This local ready signal functions as the local ready signal for the entire sequence of graphics

5   pipelines. The remaining local ready signals are provided by any graphics pipelines that are configured to work in parallel with the sequence of graphics pipelines. The global ready signal becomes asserted when each of these local ready signals is asserted. This means that the selected next pipeline, and all pipelines that are configured to work in parallel with the selected next pipeline, are ready to begin

10   outputting video data. All of the pipelines within the sequence of pipelines monitor the state of the global ready signal. When the global ready signal becomes asserted, a swap event is initiated.

By generating the local ready signal in this fashion and using the global ready signal in this fashion, the sequence of pipelines synchronizes its own internal

15   operation. The sequence of pipelines also operates in a mode that is compatible with the parallel operation of other graphics pipelines and groups of graphics pipelines.

Advantages of the invention will be set forth, in part, in the description that follows and, in part, will be understood by those skilled in the art from the description

20   herein. The advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims and equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, that are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

5   Figure 1 is a block diagram showing three graphics pipelines configured to operate in sequence.

Figure 2 is a block diagram showing the internal details of the pipeline sequence of Figure 1.

Figure 3 is a block diagram showing the pipeline sequence of Figure 1

10   operating in parallel with two graphics pipelines.

Figure 4 is a block diagram showing a single image divided into three image portions.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference will now be made in detail to preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever convenient, the same reference numbers will be used throughout the

5   drawings to refer to the same or like parts.

An embodiment of the present invention includes a method and apparatus for synchronizing the operation of graphics pipelines. This method and apparatus allows graphics pipelines to be grouped for sequential operation and allows parallel execution between these pipeline groups. In Figure 1, a parallel rendering 100

10   system is shown as a representative embodiment of the present invention. Parallel rendering system 100 is intended to represent a sequential grouping of a series of similar, or dissimilar graphics pipelines. For the specific example of Figure 1, three graphics pipelines 102a through 102c are shown. Each of these graphics pipelines 102 is intended to be representative of a wide range of different graphics pipeline

15   types. Graphics pipelines 102 are connected as a daisy chain sequence. This means that the output of each pipeline 102 (except the last pipeline 102c) is connected as an input to a subsequent pipeline 102. The output of the last pipeline 102c is connected to display device 104.

Each pipeline 102 is connected to a host processor 106. Host processor 106

20   is intended to be representative of a wide range of different processor types. The connection between host processor 106 and graphics pipelines 102 is also intended to represent a wide range of different technologies such as bus interconnects and shared memory. Host processor 106 controls the execution of pipelines 102.

Under the control of host processor 106, graphics pipelines 102 produce sequences of images for display on display 104. The production of images within pipelines 102 is synchronized. This means that graphics pipelines 102 produce images in a correct ordering (i.e., preceding images do not follow subsequent

5   images).

The internal details that provide synchronized operation are better appreciated by reference to Figure 2. In Figure 2, it may be seen that each graphics pipeline 102 includes a multiplexer 200. Multiplexer 200 has two inputs. The first of these is the digital video output of the pipeline 102 that includes the multiplexer 200.

10   The second input of each multiplexer (except the first) is connected to the output of the multiplexer 200 in a preceding pipeline 102. The second input of the first multiplexer 200 is connected to a null device.

Each pipeline 102 includes three state bits: a current bit 202, a next bit 204 and a ready disable bit 206. Current bit 202 directly controls the state of multiplexer

15   200. When current bit 202 is set, multiplexer 200 selects its first input. In this configuration, multiplexer 200 outputs the digital video output of the pipeline 102 in which it is included. For this reason, this configuration is known as local mode. When current bit 202 is not set, multiplexer 200 selects its second input. In this configuration, multiplexer 200 outputs the digital video data it receives from

20   preceding pipelines 102. For this reason, this configuration is known as pass-through mode.

Next bit 204 contains the next value for current bit 202. As will be described in more detail, pipelines 102 transfer the value included in next bit 204 to current bit

9

202 as part of each swap event. In this way, each next bit 204 controls the next state (local or pass-through) of the pipeline 102 in which it is included. For the particular embodiment being described, next bit 204 is set or cleared using a command sent by host processor 106. This allows a host application to dynamically select the next

5   pipeline 102 to output video data.

As shown in Figure 2, pipelines 102 are each connected to produce a common local ready signal 508. A pipeline 102 selected as the next pipeline 102 (i.e., a pipeline 102 having a set next bit 204) asserts local ready signal 508 when it is ready to output its video data. Host processor 106 uses ready disable bits 206 to

10   prevent all other pipelines 102 from asserting local ready signals 508. For the particular embodiment being described, ready disable bits 206 are set or cleared using a command sent by host processor 106.

Graphics pipelines 102 are connected to monitor the state of global ready signal 210. Global ready signal 210 is configured to act as a logical AND of its

15   inputs. Within the example of Figure 2 the only input for global ready signal 210 is the common local ready signal 208 produced by graphics pipelines 102. As a result, within this example, global ready signal 210 enters an asserted state whenever one of graphics pipelines 102 asserts local ready signal 208. This means that global ready signal 210 become asserted whenever a pipeline 102 having a set next bit

20   204 becomes ready to output its video data. Graphics pipelines 102 sample the state of global ready signal 210 on a periodic, synchronized basis. Typically, this is accomplished by connecting each graphics pipeline 102 to a periodically repeating vertical synchronization signal. Using this system, the synchronization subsystems

10

sample global ready signal 210 at the beginning of each pulse of the vertical synchronization signal.

When graphics pipelines 102 discover that global ready signal 210 is in the asserted state, they initiate a swap event. During the swap event, each graphics
5   pipeline 102 moves the value stored in its next bit 204 into its current bit 202. Each graphics pipeline 102 also clears its next bit 204. If the movement of next bit 204 sets current bit 202, multiplexer 200 enters local mode operation. In all other cases, multiplexer 200 either enters or stays in pass-through mode.

During operation, an application process, executing on host processor 106,
10   allocates graphics primitives to at least some of the pipelines 102. The primitives correspond to a series of images that are to be generated by pipelines 102. The host application then selects the pipeline 102 that is to render the first image portion 202. The host application then sends commands to graphics pipelines 102 to set the next bit in the selected pipeline 102. The commands also set the disable ready bits in the
15   remaining pipelines.

When the selected graphics pipeline 102 finishes rendering its graphics primitives, it asserts local swap ready 208. This causes global swap ready 210 to enter an asserted state. At the beginning of the next vertical sync pulse, the selected graphics pipelines 102 and all other graphics pipelines 102 sample global ready
20   signal 210. Since global ready signal is asserted, a swap event is initiated.

During the swap event, each graphics pipeline moves its next bit 204 to its current bit 202. In the case of the selected graphics pipeline, next bit 204 is set. This causes the selected graphics pipeline 102 to enter local mode operation. The

remaining graphics pipelines 102 have clear next bits 204. This causes these pipelines 102 to either enter or remain in pass-through mode operation.

Host processor 106 continues the rendering process by repeatedly setting appropriate next bits 204 and appropriate ready disable bits 206. Host processor

5   may also allocate new primitives to pipelines 102 that have completed rendering and outputting their video data. In this way, the present invention provides a flexible synchronization mechanism that allows a software application to selectively control the order or execution for pipelines 102.

The present invention also provides for synchronized execution between

10   sequences of graphics pipelines (such as the sequence of graphics pipelines 102 shown in Figure 2) and graphics pipelines that are configured to work in parallel with sequences of this type. An example of this type of configuration is shown in Figure 3. Figure 3, includes the three sequential pipelines 102a - 102c of Figure 2. In this case, however, two additional graphics pipelines 102d and 102e are configured for

15   parallel operation with the sequence of pipelines 102a - 102c.

For the purposes of this example, it is assumed that each that the three sequential pipelines 102a - 102c produce a series of images for a portion of display 104. The remaining pipelines 102d and 102e each provide images for additional portions of display 104. This can be appreciated by reference to Figure 4 where a

20   sample image 400 is shown. Sample image 200 is subdivided into three portions 402a through 402c. For the purposes of the example of Figure 3, it may be assumed that these portions 202 are contributed by the three sequential pipelines 102a - 102c, pipeline 102d and 102e, respectively.

12

Synchronization between the three sequential pipelines 102a - 102c and pipelines 102d and 102e is achieved by configuring pipelines 102d and 102e to provide local swap ready signals when they are ready to output their video data. This ensures that the three sequential pipelines 102a - 102c will not begin swap event

5   processing before pipelines 102d and 102e are ready. Pipelines 102d and 102e are also required to delay swap event processing until global ready signal 210 is asserted. This ensures that pipelines 102d and 102e will not begin swap event processing before the three sequential pipelines 102a - 102c are ready. In this way, the present allows pipelines grouped in sequence to operate in parallel with other

10   pipelines and groups of pipelines.

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims and equivalents.

WHAT IS CLAIMED IS:

1.   A method for synchronizing the operation of a sequence of graphics pipelines with a global ready signal, the method comprising the steps of:

3   designating, by a host processor, a next graphics pipeline within the sequence of graphics pipelines;

               waiting, by the next graphics pipeline, until the next graphics pipeline is

6   ready to output video data;

               asserting, by the next graphics pipeline, a local ready signal indicating that the next graphics pipeline is ready to output video data;

9   waiting, by the next graphics pipeline, until the global ready signal enters an asserted state; and

               outputting, by the next graphics pipeline, video data.

12   2.   A method as recited in claim 1, further comprising the step, performed by the host processor, of: disabling the assertion of local ready signals by graphics pipelines within the sequence of graphics pipelines that are not the next graphics

15   pipeline.

       3.   A method as recited in claim 1, wherein the global ready signal is formed as a logical AND of the local ready signal and other local ready signals

18   produced by other graphics pipelines.

4.   A method as recited in claim 1, wherein the global ready signal is formed as a logical AND of the local ready signal and other local ready signals

21   produced by other sequences of graphics pipelines.

5.   A method as recited in claim 1, wherein the graphics pipelines are connected to monitor the state of a periodically repeating vertical synchronization

24   signal and where the step of waiting, by the next graphics pipeline, until the global ready signal enters an asserted state further comprises the step of sampling the global synchronization at the start of each pulse of the vertical synchronization
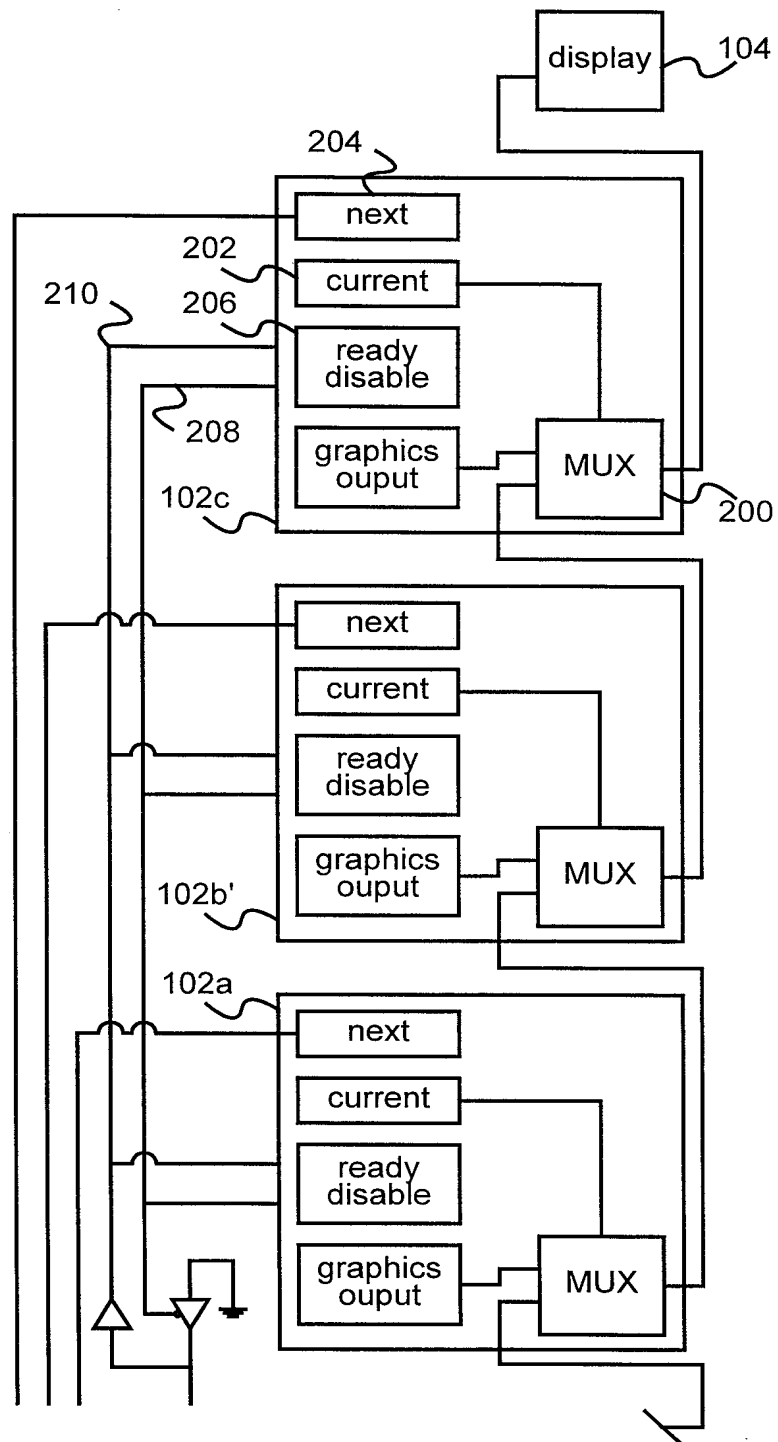
27   signal.

6.   A method as recited in claim 1, wherein each graphics pipeline in the sequence of graphics pipelines has an associated next bit, and wherein the step of

30   designating, by a host processor, a next graphics pipeline further comprises the step of setting, by the host processor, the next bit for the next graphics pipeline.
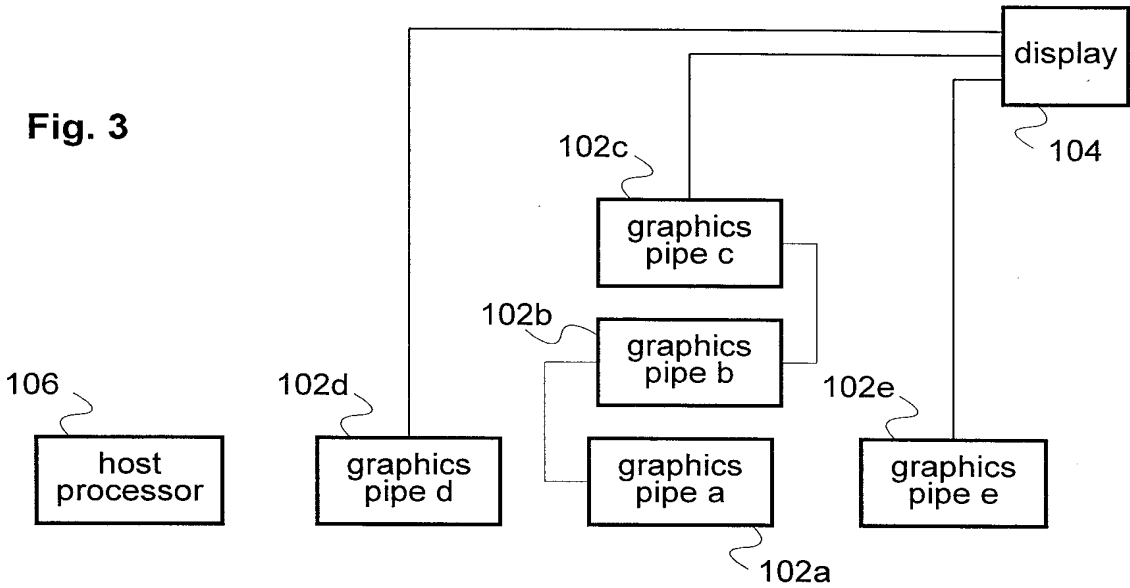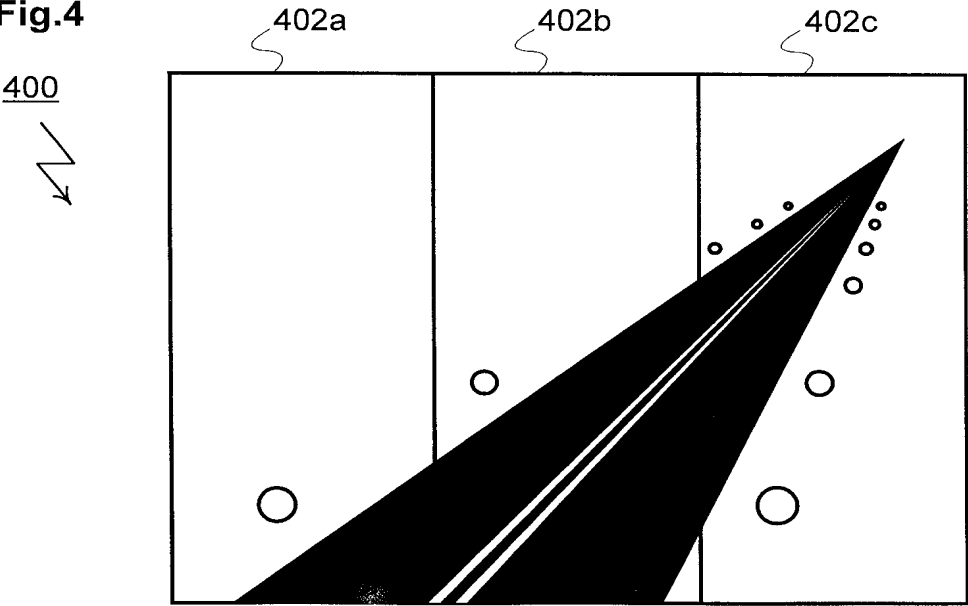
Fig. 1
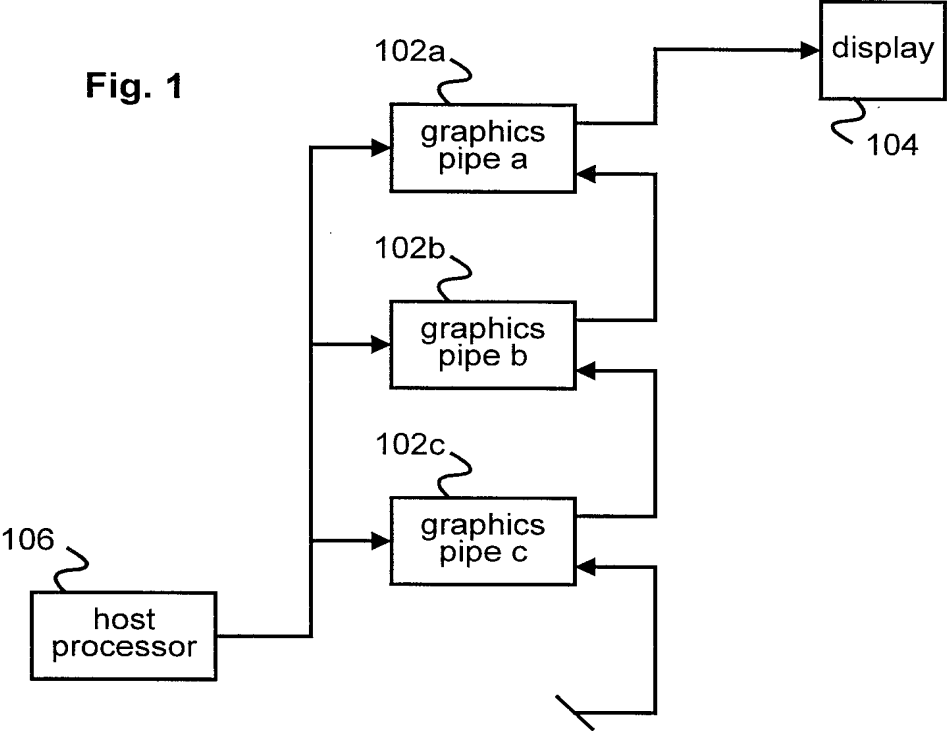
Fig. 2

Fig. 3

Fig.4

<u>400</u>

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7   G06T15/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7   G06T   G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 537 561 A (NAKAJIMA MASAITSU) 16 July 1996 (1996-07-16) column 2, line 61 -column 3, line 7 column 5, line 4 – line 31; figure 1 ---	1-6
Y	HIROYUKI SATO ET AL: "PARALLEL VISUAL COMPUTING ON THE AP1000: HARDWARE AND SOFTWARE. PARALLEL VISUAL COMPUTING ON THE AP1000: HARDWARE AND SOFTWARE" FUJITSU-SCIENTIFIC AND TECHNICAL JOURNAL,JP,FUJITSU LIMITED. KAWASAKI, vol. 29, no. 1, 21 March 1993 (1993-03-21), pages 41-49-41, XP000422963 ISSN: 0016-2523 the whole document -----	1-6

☐ Further documents are listed in the continuation of box C.   ☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 May 2000	11/05/2000

Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Authorized officer Burgaud, C

Form PCT/ISA/210 (second sheet) (July 1992)

1

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5537561	A	16-07-1996	JP	2911278 B	23-06-1999
			JP	5053803 A	05-03-1993